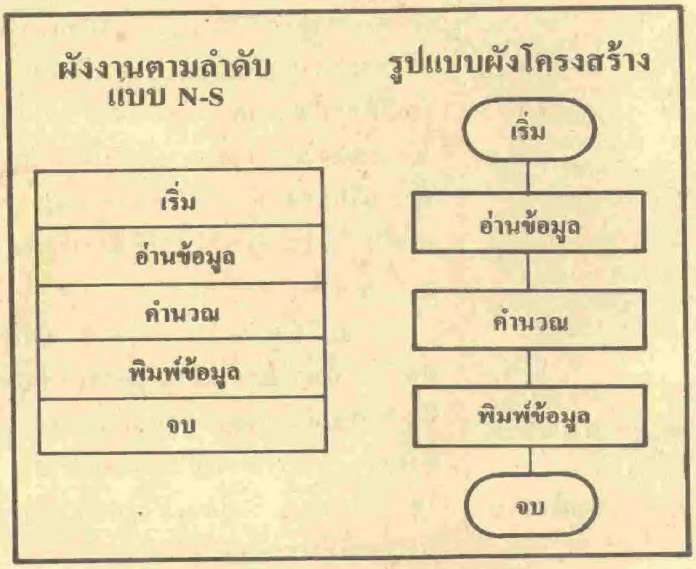
เทคนิคการออกแบบโปรแกรม ตอน 3

ผังงานแบบ Nassi-Shneiderman

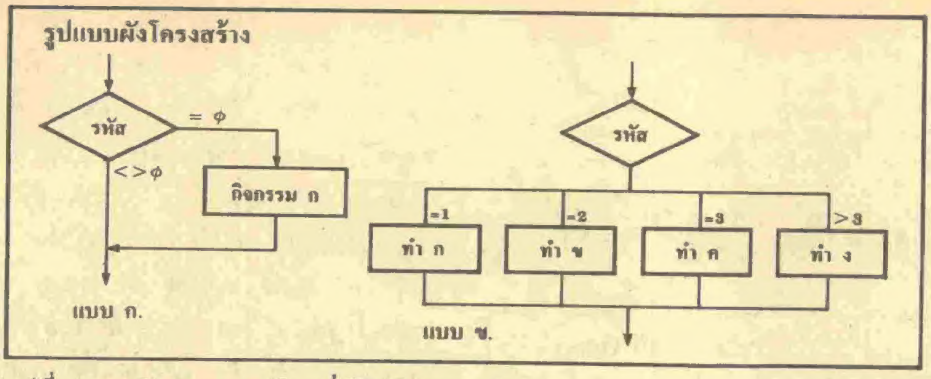
อาตาปี & VIP

หลักการแบ่งแยกและลำดับชั้น

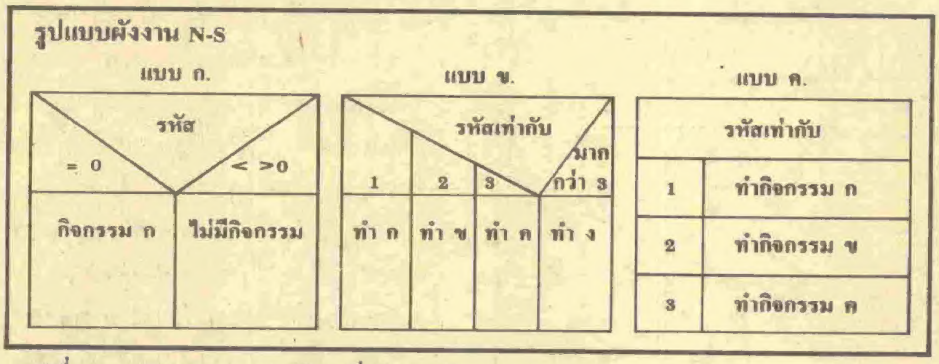
กลุ่มแนวความคิดแบบโครงสร้างได้ค้นหาวิธีการที่จะปรับปรุงผังงานตามปรัชญาของพวกตนให้เห็นภาพลักษณ์ที่เข้าใจง่าย ได้มีการค้นคิดเสนอรูปต่าง ๆ จนมีหลายแบบต่างกันไป แต่ก็ยังคงควบคุมรูปแบบพื้นฐานทั้งสาม คือตามลำดับเลือกและทำซ้ำ ดังเช่นแผนงานแบบ Nassi-Shneiderman หรือที่เรียกย่อ ๆ ว่า ผังงาน N-S ซึ่งตั้งตามชื่อผู้คิดค้นทั้งสองคนนั้น เป็นภาพสะท้อนของปรัชญาโครงสร้างด้านหลักการแบ่งแยกและลำดับชั้นได้เป็นอย่างดี อย่างเช่น รูปแบบ N-S แปลกกว่าผังงานทั่วไป ไม่มีลูกศรกำหนดทิศทาง สัญลักษณ์ที่ใช้มีแต่เส้น แบ่งกิจกรรมเป็นโมดูล จัดเป็นชั้น ๆ อันเป็นลักษณะของเทคนิคการออกแบบจากบนลงล่าง จากกล่องใหญ่แบ่งออกเป็นกล่องย่อย ๆ ซึ่งเป็นตัวแทนของกิจกรรม กล่องใหญ่ควรแบ่งซอยออกเป็นกล่องย่อยระหว่าง 10 ถึง 20 หากมากกว่านั้นให้แยกกล่องออกมาเป็นชุดใหม่ ด้วยวิธีแบ่งเป็นกล่องกิจกรรม เช่นนี้จึงทำให้เห็นภาพของลักษณะความสัมพันธ์ระหว่างกิจกรรมที่เป็นกิจกรรมซ้ำหรือซ้อนได้ชัดเจนกว่าผังงานแบบอื่น อย่างไรก็ตามเมื่อมีข้อดีก็มีข้อเสีย ผังงาน N-S ไม่มีการใช้การข้ามกิจกรรมแบบ GOTO หรือ branch อย่างไม่มี



รูปที่ 1 แสดงรูปแบบผังงาน N-S เทียบกับผังงานโครงสร้าง



รูปที่ 2.1 ผังงานแบบเลือกที่เขียนในแบบต่าง ๆ



รูปที่ 2.2 ผังงานแบบเลือกที่เขียนได้แบบต่าง ๆ กัน

เงื่อนไข ดังนั้นจึงขาดการแสดงความสัมพันธ์ชนิดแบบข้ามกระโดด ซึ่งมีระหว่างข้อมูลด้วยกัน หรือระหว่างข้อมูลกับกิจกรรมต่าง ๆ อันเป็นสิ่งจำเป็นของการออกแบบฐานข้อมูล ดังนั้นจึงมีผู้นิยมใช้ผังงานแบบนี้ไปในการออกแบบเกี่ยวกับลอจิกของโปรแกรมที่บอกเฉพาะขั้นตอนการทำงาน หากนำไปใช้ออกแบบฐานข้อมูลก็ใช้ร่วมกับเครื่องมืออย่างอื่น ซึ่งจะกล่าวถึงโอกาสต่อไป รูปแบบของผังงาน N-S เมื่อเทียบกับผังงาน โครง–สร้าง และการประยุกต์ใช้มีดังรูปที่ 1 เป็นผังงานตามลำดับ

ตัวอย่างการใช้งานกับภาษาเบสิค

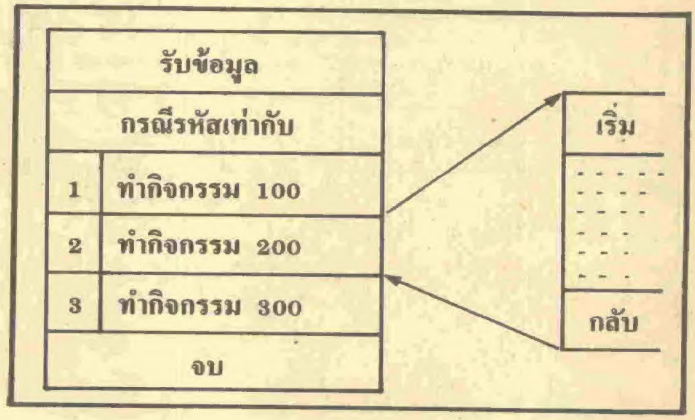
```
10 rem start program
20 input a
30 c=(a*.04)/2
40 print a,c
50 stop
```

ผังงานแบบเลือก เขียนได้หลายแบบดังรูปที่ 2.1 และ 2.2

ตัวอย่างการใช้งานผังงานกับภาษาต่าง ๆ ดังรูปที่ 3, 4

ภาษาเบสิก

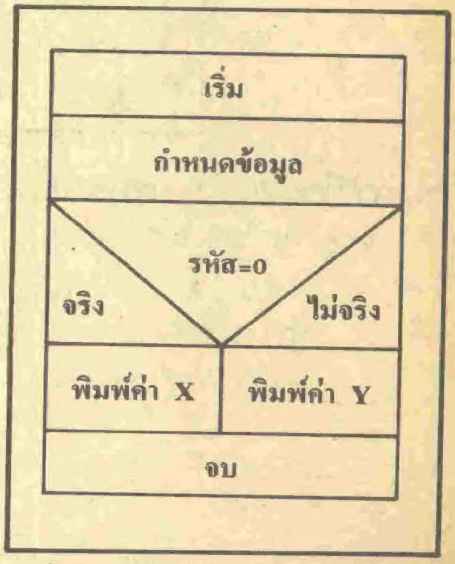
```
10  input a
20  on a gosub 100,200,300
30  stop
100 rem sub 100
110 return
200 rem sub200
210 return
300 rem sub300
310 return
```



รูปที่ 3 การใช้ผังงานกับภาษาเบสิก เริ่มโปรแกรมย่อยกลับ

ภาษาฟอร์แทรน

```
100 READ (15,900) A,B
    X = A*3.0
    Y = 10.0*B
    Z = A + B
    IF (Z . EQ. 0.0) GO TO 200
    WRITE (16,910) Y
    GO TO 300
200 WRITE (16,910) X
300 STOP
```



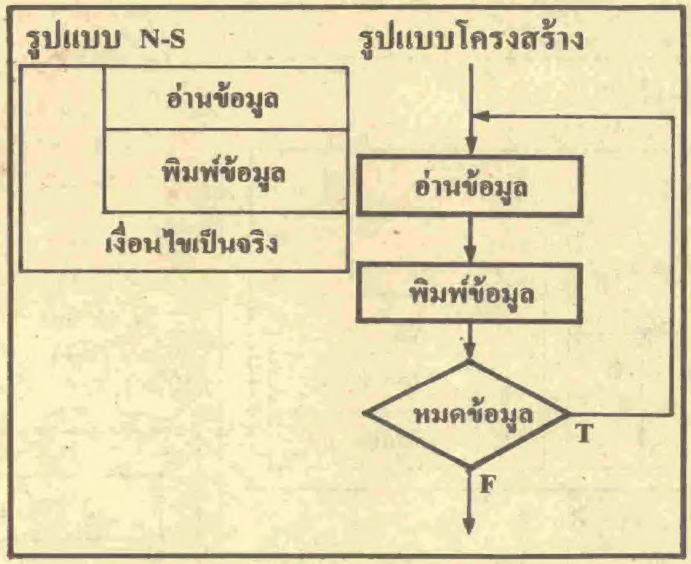
รูปที่ 4 การใช้ผังงานกับภาษาฟอร์แทรน

ผังงานทำซ้ำ มี 2 ประเภทคือ

1. ทำซ้ำจนกระทั่ง (DO-UNTIL)

ตัวอย่างการใช้งานผังงานกับภาษาโคบอล

```
READ RECORD AT END MOVE "Y" TO DONE-PROCESSING.
PERFORM PROCESS - DATA UNTIL DONE-PROCESSING IS EQUAL TO "Y".
STOP RUN.
```



รูปที่ 5 แสดงผังงานทำซ้ำจนกระทั่งที่ใช้กับภาษาโคบอล

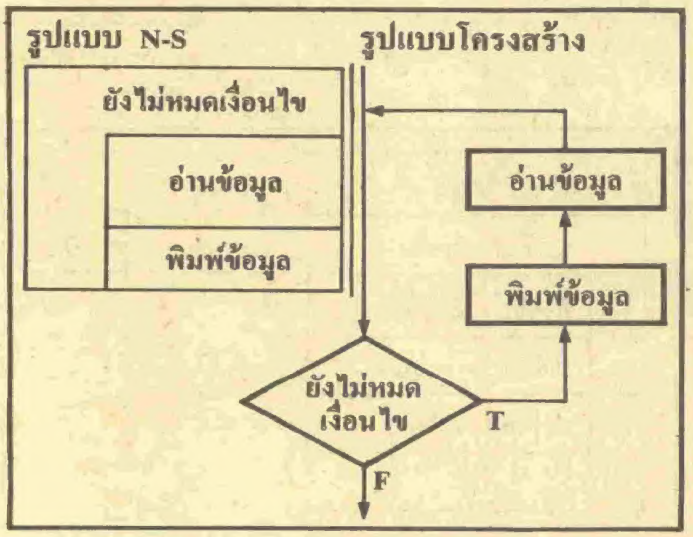
2. ทำซ้ำในขณะที่ (DO-WHILE)

ตัวอย่างการใช้งานกับภาษาเบสิก

```
FOR I=K TO L
    INPUT A
    PRINT A
NEXT I
```

การออกแบบโปรแกรมด้วยผังงาน N-S

มีหลักการคล้ายการออกแบบผังงานแบบโครงสร้างด้วยการใช้หลักการแบ่งแยกและลำดับชั้น เริ่มต้นด้วยการมองกิจกรรมเป็นกล่องดำขนาดใหญ่ แล้วค่อย ๆ แยกกิจกรรมออกไปเป็นกิจกรรมย่อย ดังตัวอย่างงานต่อไปนี้



รูปที่ 6 แสดงผังงานทำซ้ำในขณะที่ใช้กับภาษาเบสิก

ลักษณะเป็นงานการสร้าง แก้ไขข้อมูลและเรียกดูข้อมูลทะเบียนผู้สอบบัญชีที่มีจำนวนประมาณพันกว่ารายข้อมูลขนาดนี้ใช้เครื่องไมโครคอมพิวเตอร์ด้วยโปรแกรมดีเบส ทำได้อย่างมีประสิทธิภาพ ปัญหามีอยู่ว่าจะออกแบบโปรแกรมให้ดีต้องทำอย่างไร คำตอบคือ ใช้เครื่องมือและขั้นตอนการออกแบบที่เคยกล่าวมาแล้ว ในที่นี้จะใช้เครื่องมือผังงาน N-S มาช่วย ก่อนอื่นพิจารณาความต้องการของผู้ใช้เพื่อแบ่งกิจกรรมที่ต้องการเป็นโมดูลกว้าง ๆ พิจารณาถึงกิจกรรมที่มีความแตกต่าง จัดแบ่งเป็นกลุ่มกิจกรรม การรวบรวมจัดแยกกิจกรรมได้ด้วยการสอบถามผู้ใช้ในเรื่องแหล่งที่มาของข้อมูล ชนิดของข้อมูล การเก็บข้อมูล การนำข้อมูลออกมาใช้ และการบำรุงรักษาข้อมูลตลอดจนถึงการปรับปรุงและเพิ่มข้อมูลให้ทันสมัย จากความต้องการของผู้ใช้วิเคราะห์จำแนกออกได้เป็น 5 กิจกรรมใหญ่ ๆ คือ

กิจกรรมที่ 1. *สร้างหรือเพิ่มข้อมูล*
กิจกรรมที่ 2. *แก้ไขข้อมูลที่มีอยู่*
กิจกรรมที่ 3. *ลบข้อมูล*
กิจกรรมที่ 4. *เรียกดูข้อมูลทางจอภาพ*
กิจกรรมที่ 5. *พิมพ์รายงานแบบต่าง ๆ*

จากกิจกรรมทั้ง 5 บวกด้วยกิจกรรมที่เป็นระเบียบของการดำเนินงานของโปรแกรมอีก 2 กิจกรรมคือ กิจกรรมเริ่มต้น และกิจกรรมจบ อันเป็นการเตรียมการและเตรียมจบ ในที่นี้ผู้ใช้ต้องการเรียกดูและทำงานด้วยเครื่องไมโครคอมพิวเตอร์ ลักษณะแบบโต้ตอบโดยให้แสดงผลที่จอภาพ การออกแบบโปรแกรมจึงต้องคำนึงถึงความสะดวกในการใช้งานด้วย ดังนั้น กิจกรรมเริ่มต้นแตกเป็นกิจกรรมย่อย ๆ ด้วยการจัดทำเป็นแบบสารบัญแบบเมนูอาหารให้ผู้ใช้เลือกเลขรหัสกิจกรรม ต่อจากนั้นก็แยกเป็นกิจกรรมต่าง ๆ ผังงานเป็นรูปที่ 7

```
เริ่มโปรแกรม
กำหนดเงื่อนไขและรหัสสารบัญ
ทำกิจกรรมต่าง ๆ ในขณะที่
ยังไม่หมดเงื่อนไข
    สารบัญกิจกรรมหลัก000
    เลือกรหัสกิจกรรม
    ทำกรณีรหัสกิจกรรมเท่ากับ
        1. สร้างหรือเพิ่มข้อมูล        100
        2. แก้ไขข้อมูลที่มีอยู่         200
        3. ลบข้อมูล                  300
        4. เรียกดูข้อมูลจากจอภาพ     400
        5. พิมพ์รายงานแบบต่าง ๆ      500
        6. จบกิจกรรม                 600
หมดเงื่อนไข
จบโปรแกรม
```

รูปที่ 7 การนำผังงาน N-S มาช่วยในการออกแบบโปรแกรม

ในขั้นตอนนี้เท่ากับเป็นการทำโครงสร้างหลักผังงานแบบ N-S เรียกว่ากล่องหลัก ขั้นต่อไปต้องแจงออกเป็นกล่องย่อย ๆ อีกหลายชุด บางชุดอาจมีขนาดใหญ่กว่ากล่องหลักเสียอีกขนาดของกล่องวัดจากจำนวนคำสั่งหรือกิจกรรม (อย่าลืมข้อกำหนดขนาดของกล่อง-โมดูล ควรอยู่ระหว่าง 10 ถึง 20 กิจกรรม) ดังนั้นแต่ละชุดของกล่องควรมีรหัสประจำชุด สำหรับชุดหลักให้รหัสว่า 000 ลองแปลผังงานของกล่องหลักเป็นคำสั่งซึ่งทำได้คล้ายดังรูปที่ 8

จากนั้นก็วิเคราะห์แต่ละชุดของกล่องกิจกรรมต่าง ๆ แยกย่อยลงไปพร้อมกับทำการสังเคราะห์คือ การทำความสัมพันธ์และการจัดลำดับชั้นระหว่างชุดกล่องต่าง ๆ ในการแยกย่อยแต่ละครั้งให้คำนึงถึงภาพของคำสั่งหรือภาษาโปรแกรมไปด้วย เพราะหากเมื่อไรกิจกรรมย่อยนั้นแปลงเป็นภาษาคำสั่งได้แสดงว่ากิจกรรมนั้นไม่ต้องแบ่งตัวแยกย่อยอีกแล้ว ปกติภาพของกิจกรรมที่เป็นคำสั่งได้มักจะเป็นการแตกย่อยออกมาในขั้นที่ 2 หรือ 3 ในที่นี้คือ ใช้โปรแกรมดีเบสทรีเขียน ขอยกตัวอย่างบางกิจกรรมคือ กิจกรรมรหัสที่ 2 การแก้ไขข้อมูล เริ่มต้นด้วยแยกเป็นกล่องชุดใหม่ ให้รหัสประจำกล่องเป็น 200 วิเคราะห์ดูว่าการแก้ไขข้อมูลแบ่งเป็นกิจกรรมหยาบ ๆ ออกเป็นกลุ่มก้อนที่แตกต่างกันได้หรือไม่ โดยดูจากประเภทข้อมูลหรือที่เรียกกันว่าแฟ้มข้อมูล ในตัวอย่างนี้สำรวจแล้วแบ่งออกได้เป็น 4 แฟ้ม ดังนั้นกิจกรรมแก้ไขข้อมูลจึงมี 4 กิจกรรมหยาบ ๆ และก่อนที่จะทำการแยกกล่องออกมาต้องทำการสังเคราะห์เสียก่อนว่าจะสร้างความสัมพันธ์เพื่อกลับไปกล่องหลักอย่างไร ในที่นี้ดูว่าจากกล่องหลักมาเรียกกล่องย่อยด้วยการเลือกกิจกรรม เมื่อทำกิจกรรมกล่องย่อยเสร็จก็

```
* = = = = = = = = = = = = = = = = = = => main 000
set talk off
clear
store " " to t
store " " to ch
do while t=" "
    @ 3,30 say "   สารบัญ"
    @ 6,28 say "1. เพิ่มข้อมูล"
    @ 8,28 say 2. แก้ไขข้อมูล"
    @ 10,28 say "3. ลบข้อมูล"
    @ 12,28 say "4. เรียกดูข้อมูลทางจอภาพ"
    @ 14,28 say "5. พิมพ์รายงาน"
    @ 16,28 say "6. จบงาน"
    @ 19,30 say "ใส่หมายเลขที่ต้องการ >>" get ch
    read
    do case
        case ch= "1"
          do act 100
        .
        .
        case ch="6"
          use

          quit
        otherwise
        loop
    endcase
enddo
```

รูปที่ 8 แสดงการแปลผังงานของกล่องหลักเป็นคำสั่ง

รูปที่ 9 ผังงานแสดงการทำความสัมพันธ์ระหว่างชุดกล่องต่าง ๆ

เริ่มกิจกรรม 200			
กำหนดเงื่อนไขและรหัสสารบัญ			
ทำในขณะที่เงื่อนไขเป็นจริง			
	สารบัญย่อย 200		
	เลือกรหัสกิจกรรม		
	1.	แก้ไขแฟ้มผู้สอบบัญชี	210
	2.	แก้ไขแฟ้มใบนำส่ง	220
	3.	แก้ไขแฟ้มข้อมูลสถิติ	230
	4.	แก้ไขแฟ้มข้อมูลสถานะภาพ	240
	5.	กลับไปที่สารบัญกิจกรรมหลัก	000
หมดเงื่อนไขกลับไปที่กิจกรรม			000

รูปที่ 10 ผังงานแสดงการแก้ไขข้อมูลสถานะ

เริ่มกิจกรรม 240			
กำหนดเงื่อนไข			
ทำในขณะที่เงื่อนไขเป็นจริง			
	ล้างจอภาพ		
	กำหนดรหัสสถานะ		
	เปิดแฟ้มข้อมูลสถานะ		
	เทียบรหัส		
	ถ้าค้นหมดแฟ้มแล้วไม่พบ		
	ไม่พบ	แสดงข้อความ "ข้อมูลไม่มี"	
	พบ	แสดงข้อมูลสถานะ	
	ถ้าต้องการแก้ไขข้อมูลอื่นอีก		
	แก้	กลับไปใส่รหัสค้นหาใหม่	
	ไม่แก้	เปลี่ยนเงื่อนไขเพื่อจบกิจกรรม	
ตรวจเงื่อนไข			
ปิดแฟ้ม			
กลับไปกิจกรรม 200			

ควรกลับไปที่กิจกรรมกล่องหลัก ดังนั้นจึงต้องเพิ่มกิจกรรมสำหรับกล่องชุดที่ 200 อีกหนึ่งกิจกรรมคือการกลับไปสารบัญกิจกรรมหลัก 000 เขียนเป็นผังงานได้ดังรูปที่ 9

ขั้นต่อไปแยกกิจกรรมออกเป็นชุดย่อย ๆ ต่อจากกล่องชุดที่ 200 วิเคราะห์แยกละเอียดลงไปถึงระดับเนื้อหาของข้อมูลแต่ละแฟ้มหรือที่เรียกว่าพิจารณาแยกเป็นชนิดของข้อมูลรายประเภท หรือที่เรียกว่าข้อมูลระดับระเบียน ก็จะได้ภาพของกิจกรรมการแก้ไขและวิธีการแก้ไขข้อมูลแบบต่าง ๆ ขั้นต่อไป วิเคราะห์จากกิจกรรมระดับข้อมูลลงเป็นระดับพอที่จะกำหนดเป็นคำสั่งได้ ซึ่งจะเป็นการแยกย่อยสุดท้าย ในที่นี้ขอยกตัวอย่างเฉพาะกิจกรรมที่ 240

ซึ่งเป็นการแก้ไขข้อมูลสถานะ มีรหัสสถานะเป็นตัวหลักในการค้นแฟ้มข้อมูลสถานะ เมื่อเรียกรหัสใดแล้วให้แสดงรายละเอียดสถานะที่เก็บเป็นแต่ละระเบียน สามารถแก้ข้อความส่วนใดของข้อมูลก็ได้ บางครั้งอาจแก้ติดต่อกันหลายระเบียบจนพอใจแล้วจึงกลับไปที่กิจกรรมย่อย 200 จากผังงานได้ดังรูปที่ 10 สามารถนำไปเขียนเป็นโปรแกรมได้

ผังงาน N-S เป็นเครื่องมือที่ช่วยในการออกแบบในรายละเอียดของโปรแกรมที่ดีผังงานหนึ่ง เมื่อออกแบบผังงานด้วยวิธีถอดกล่องออกเป็นชั้น ๆ ในแต่ละชั้นก็นึกภาพภาษาไปด้วยจะพบว่าหากใช้ภาษาที่มีลักษณะเป็นโครงสร้างอยู่แล้วเช่นภาษาปาสคาล (หรือแม้แต่โปรแกรมสำเร็จรูปดีเบส ก็นับเนื่องเป็นภาษาที่มีรูปแบบโครงสร้างได้) สามารถออกแบบจากความต้องการที่เป็นนามธรรมแยกเป็นกล่องย่อยจนสามารถกำหนดกิจกรรมเป็นคำสั่งในภาษาแทนกิจกรรมนั้นได้